Amendments to the Drawings:

The one (1) annotated sheet of drawing includes changes to Figure 4. These changes consist of adding replacing the reference numbers "411," "421," "423," "425," and "427," with the reference numbers "401," "403," "405," "407," and "409," respectively.

Attachment: One (1) annotated sheet corresponding to Figure 4.

REMARKS/ARGUMENTS

This Amendment is in response to the Office Action dated April 18, 2006. Claims 1, 2 and 4-12 are pending in the present application. Claims 1, 2 and 4-12 have been rejected. Claims 1, 8, and 11 have been amended to address objections, a §101 rejection, and a §112 rejection. Applicants respectfully submit that no new matter has been presented. For the reasons set forth more fully below, Applicants respectfully submit that the claims as presented are allowable. Consequently, reconsideration, allowance, and passage to issue are respectfully requested.

Drawings

The Examiner has stated:

> **The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they do not include the following reference sign(s) mentioned in the description: 401, 403, 407 and 409 (see page 24, lines 22-28, of the specification)...**

In response, Figure 4 has been amended such that the reference numbers "411," "421," "423," "425," and "427" have been replaced with the reference numbers "401," "403," "405," "407," and "409," respectively. The corrected drawing is attached hereto.

Claim Objections

The Examiner has stated:

> **Claims 1, 2 and 4 are objected to because of the following informalities: Independent claim 1, in step e), states that there is at least one connector between the end user application and the application server. This limitation is inherent as a result of the added limitation in step b) (the web server comprises a connector), so it is not clear if step e) refers to at least one additional connector or if this limitation will be satisfied if step b) is satisfied. Further, it is not clear if steps (i) and (ii) are performed by the at least one connector. Appropriate correction is required.**

> **Claims 8-10 are objected to because of the following informalities: there is a lack of antecedent basis for "the client" in line 3 of claim 8. Also, it is not clear if the connector is supposed to receive the application request from the server (see claim 8, part C line 1) or if Applicant intended for the connector to receive a response from the server. Claims 9 and 10 inherit these deficiencies. Appropriate action is required.**

In response, claims 1 and 8 have been amended to address the above-referenced objections. Specifically, claim 1 has been amended to clarify that the connector performs steps (i) and (ii). Also, claim 8 has been amended to provide antecedent basis for the term "client" on line 3 and for the term "response" in step c), and to clarify in step c) that the connector receives the response. Dependent claims 2, 4, 9, and 10 depend from claims 1 and 8, respectively. Accordingly, Applicants respectfully submit that claims 2, 4, 9, and 10 overcome the objections for at least the same reasons as claims 1 and 8.

Claim Rejections - 35 U.S.C. §101

The Examiner has stated:

> **Claims 11 and 12 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The language of claim 11 raises questions as to whether computer hardware is required to implement the claimed invention. Claim 11 specifies that the program product comprises a storage medium. Paper on which computer instructions can be printed is a storage medium and is not statutory because it is not considered to be descriptive matter. Therefore, "storage medium," as opposed to "computer readable storage medium," is non-statutory.**

In response, claim 11 has been amended to clarify that the storage medium is a computer-readable storage medium. Applicants respectfully submit that claim 11, as amended, now complies with 35 U.S.C. §101. Dependent claim 12 depends from claim 11. Accordingly, Applicants respectfully submit that claim 12 overcomes the rejection for at least the same reasons as claim 11.

Claim Rejections - 35 U.S.C. §112

The Examiner has stated:

> **Claims 8-10 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Claim 8 states "an application request" in both lines 4 and 6. It is not clear which request is referred to by "the application request" throughout the claim. Claims 9 and 10 inherit this deficiency.**

In response, claim 8 has been amended to address the above-referenced rejection. Specifically, the second instance of the phrase "an application request" has been corrected to read "the application request". Applicants respectfully submit that claim 8, as amended, now complies with 35 U.S.C. §112, second paragraph. Dependent claims 9 and 10 depend from claim 8, respectively. Accordingly, Applicants respectfully submit that claims 9 and 10 overcome the rejection for at least the same reasons as claim 8.

Claim Rejections - 35 U.S.C. §103

The Examiner has stated:

> **Claims 1-2 and 4-12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ben-Shachar (US 5,761,656), in view of Flanagan et al. (US 6,243,737 B1).**
>
> **As to claim 1, Ben-Shachar substantially discloses a method of processing an application request on an end user application and an application server (abstract) comprising the steps of:**
>
> **a) initiating the application request on the end user application in a first language with a first application program (col. 5 lines 6-10);**
>
> **b) transmitting the application request to the server and converting the application request from the first language of the first end user application to a form for the language running on the application server (col. 5 lines 3-12), wherein the end user application is connected to the application server through a web server, and the web server comprises a connector (col. 6 lines 18-32);**
>
> **c) processing said application request on the application server (col. 5 lines 6-15);**
>
> **d) transmitting a response to the application request from the application server to the end user application, and converting the response to the application request from the language running on the application server to the first language of the first end user application (col. 5 lines 12-17); and**

> **e) wherein the end user application and the application server have at least one connector therebetween (col. 5 lines 6-15), and the steps of (i) converting the application request from the first language of the first end user application as a source language to the language running on the application server as a target language (col. 5 lines 6-12), and (ii) converting a response to the application request from the language running on the application server as a source language to the first language of the first end user application as a target language (col. 5 lines 13-16), each comprise the steps of:**
>
> **1) invoking connector metamodels of respective source language and target language ("mapping file" col. 5 lines 6-15);**
>
> **2) populating the connector metamodels with metamodel data of each of the respective source language and target language, the metamodel data of the target language including a map, mapset, and a mapfield (Figure 3; col. 5 lines 29-48; col. 9 lines 10-22); and**
>
> **3) converting the source language to the target language (col. 9 lines 10-22).**
>
> **Ben-Shachar fails to specifically disclose a mapping support language. However, Flanagan et al. substantially disclose a mapping support language (col. 10 lines 1-16). It would have been obvious to one of ordinary skill in the art at the time of Applicant's invention to combine these references because both references focus on providing server processing to clients...**

Applicants respectfully disagree with the Examiner's rejections. The present invention provides a method of processing an application request on an end user application and an application server including a mapping support language. In accordance with the present invention, the method includes: a) initiating the application request on the end user application in a first language with a first application program; b) transmitting the application request to the server and converting the application request from the first language of the first end user application to a form for the mapping support language running on the application server, wherein the end user application is connected to the application server through a web server, and the web server comprises a connector; c) processing said application request on the application server; and d) transmitting a response to the application request from the application server to the end user application, and converting the response to the application request from the mapping support language running on the application server to the first language of the first end user application. The connector is configured to (i) convert the application request from the first language of the first end user application as a source language to the language running on the application server as a target language, and (ii) convert a response to the application request from the language running on the application server as a source language to the first language of the first end user application as a target language. Each includes the steps of: 1) invoking connector metamodels of respective source language and target mapping support language; 2) populating the connector metamodels with metamodel data of each of the respective source language and target mapping support language, the metamodel data of the target mapping support language including a map, a mapset, and a mapfield; and 3) converting the source language to the mapping support language. Ben-Shachar in view of Flanagan does not teach or suggest these features, as discussed below.

Ben-Shachar discloses method interacting between databases and graphical user interfaces. To provide an interface between a computer application and a database, a user invokes the application and another program, called "link manager." The user "drags and drops" GUI objects from the application window to the link manager window to command the link manager to associate parameters of the GUI objects (e.g., object names) with input and output fields of database operations. The link manager records these associations in a mapping file. When later the application is executed, a program, called "execution manager," reads the mapping file. When the application issues a request to the database, the execution manager uses the mapping file to translate the request to the database format, converting values of parameters of GUI objects to input values of a database operation as defined by the mapping file. The execution manager translates the database response to the application format, converting output values of the response to values of application GUI parameters as defined by the mapping file. (Abstract.)

Ben-Shachar does not teach or suggest "converting the application request from the first language of the first end user application to a form for the mapping support language running on the application server," as recited in independent claims 1, 5, and 8. The Examiner has referred to column 5, lines 3-12, of Ben-Shachar as teaching converting an application request to a form for a language running on an application server. However, Ben-Shachar specifically states at column 5, lines 6-11, that an execution manager "constructs mapping objects which translate the application requests into formats of database 130. The DB 130 formats are defined by the DB API." The translation into **formats of a database** where the formats are specifically **defined by the database application program interface**, as taught by Ben-Shachar, is clearly different from the conversion "to a form for the mapping support language **running on the application server**" as in the present invention.

Applicants agree with the Examiner that Ben-Shachar fails to specifically disclose a mapping support language. The Examiner has relied on Flanagan to cure the defects of Ben-Shachar. Flanagan discloses a method, computer-readable medium, and system for providing direct transaction access to information residing on a host system. The system includes a transaction server that routes transactions between the host system and a client device. The transaction server maps client transactions to host transactions using direct transaction access that allows the transaction server to bypass the displays of the host system and to obtain information directly from the host source. Each client transaction is mapped to one or more host transactions. Each host transaction includes inputs and outputs. The host inputs are resolved to addressable references before run-time. The transaction server processes the client transactions concurrently. A graphical user interface is provided that allows a user to create client transactions without writing any programming code. The computer readable-media is loaded on the transaction server without requiring additional software on the host system. (Abstract.)

The Examiner has referred to column 10, lines 1-16, of Flanagan as disclosing the mapping support language. However, nowhere does this section of Flanagan specifically teach or suggest a mapping support language. Instead, Flanagan describes a "host transaction map," which is merely a "file that defines the input and output display data for a particular host transaction" (column 10, lines 5-7). Combining Flanagan with Ben-Shachar would not provide the present invention, but would instead provide a method for translating an application request into the format of a host transaction map, which would not make sense if implemented in the present invention.

Since neither Ben-Shachar nor Flanagan teach or suggest the mapping support language of the present invention, Ben-Shachar in view of Flanagan does not teach or suggest the *combination of* "converting the response to the application request from the **mapping support language** running on the application server to the first language of the first end user application" and "populating the connector metamodels with metamodel data of each of the respective source language and target mapping support language, **the metamodel data of the target mapping support language including a map, a mapset, and a mapfield,**" as recited in independent claims 1, 5, and 8. The Examiner has referred to Figure 3, column 5, lines 29-48, and column 9, lines 10-22, of Ben-Shachar as having aspects of these features. However, Figure 3 merely shows "mappers" and column 5, lines 29-48, states that the mappers map data between data between "self-describing object fields" and databases, HTML viewers and PDF applications. Column 9, lines 10-22, of Flanagan merely describe "map fields" in the context of "mapping from an HTML viewer 120 to database 130" (see column 9, lines 6-9). Combining the mapping elements of Ben-Shachar with the "host transaction map" of Flanagan clearly fails to teach or suggest "mapping support language" and "the metamodel data of the target mapping support language including a map, a mapset, and a mapfield," of the present invention.

Therefore, Ben-Shachar in view of Flanagan does not teach or suggest the cooperation of elements as recited in independent claims 1, 5, and 8, and these claims are allowable over Ben-Shachar in view of Flanagan.

Independent claim 11

Similar to independent claims 1, 5, and 8, independent claim 11 recites "mapping support language metamodel data including a map, a mapset, and a mapfield." As described above, with respect to independent claims 1, 5, and 8, Ben-Shachar in view of Flanagan does not teach or suggest these elements. Accordingly, the above-articulated arguments related to independent claims 1, 5, and 8 apply with equal force to claim 11. Therefore, claim 11 is allowable over Ben-Shachar in view of Flanagan for at least the same reasons as claims 1, 5, and 8.

Dependent claims

Dependent claims 2, 4, 6-7, 9-10, and 12 depend from independent claims 1, 5, 8, and 11, respectively. Accordingly, the above-articulated arguments related to independent claims 1, 5, 8, and 11 apply with equal force to claims 2, 4, 6-7, 9-10, and 12, which are thus allowable over the cited references for at least the same reasons as claims 1, 5, 8, and 11.

Conclusion

In view of the foregoing, Applicants submit that claims 1, 2, and 4-12 are patentable over the cited references. Applicants, therefore, respectfully request reconsideration and allowance of the claims as now presented.

Applicants' attorney believes that this application is in condition for allowance. Should any unresolved issues remain, the Examiner is invited to call Applicants' attorney at the telephone number indicated below.

Respectfully submitted,
SAWYER LAW GROUP LLP

July 17, 2006
Date

Joseph A. Sawyer, Jr.
Attorney for Applicant(s)
Reg. No. 30,801
(650) 493-4540

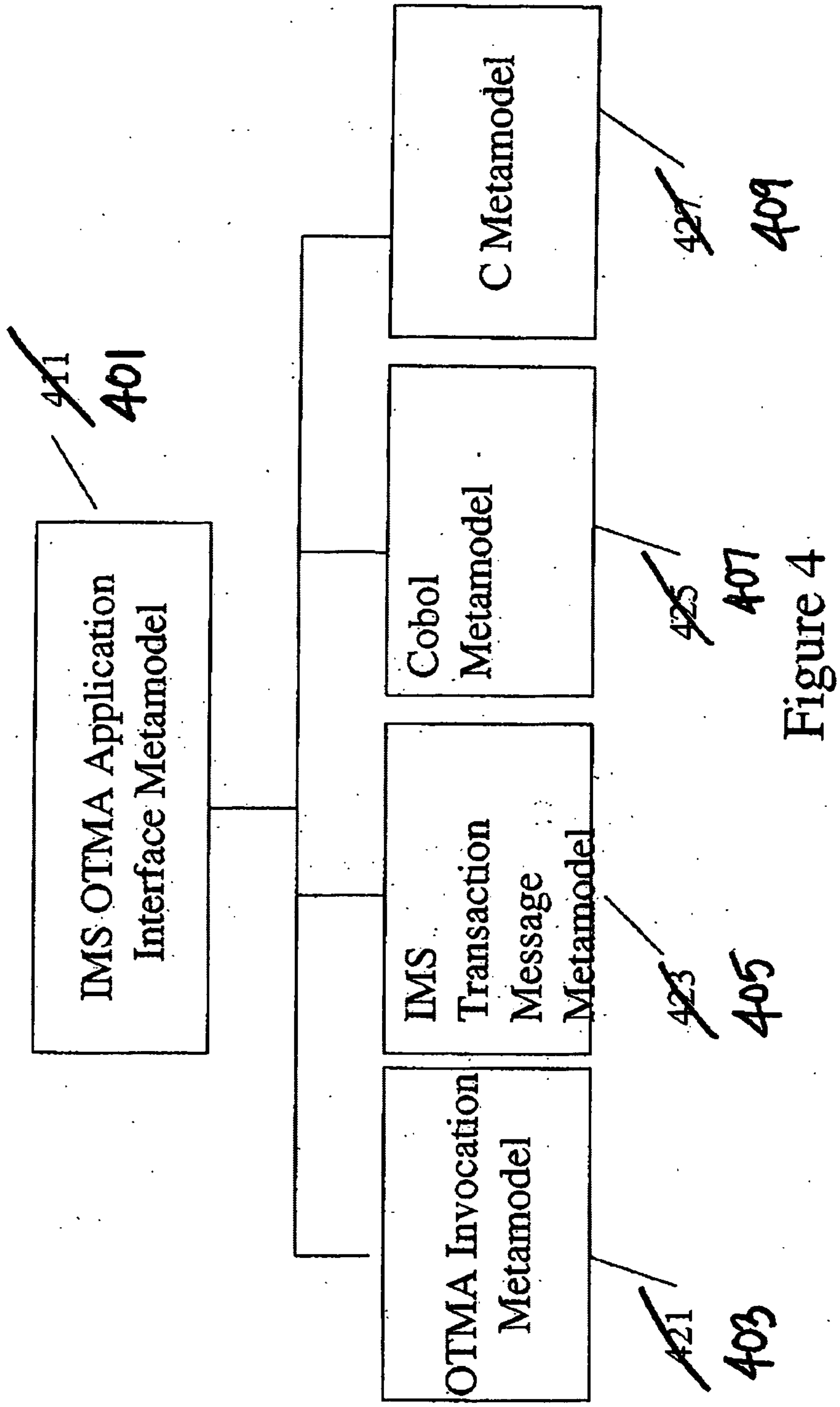
IMS OTMA Application Interface Metamodel
~~411~~
401
OTMA Invocation Metamodel
~~421~~
403
IMS Transaction Message Metamodel
~~423~~
405
Cobol Metamodel
~~425~~
407
C Metamodel
~~427~~
409

Figure 4